**SOS2 constraints in GLPK**

Typeset from an ASCII posting to <help-glpk@gnu.org> from Andrew Makhorin on 02 June 2007. Additional note provided on 22 December 2008. Minor changes made to the text. Prepared by Robbie Morrison <robbie@actrix.co.nz> 23 December 2008. Release 02.

**Additional note**: piecewise linear functions expressed through SOS2 may be used to model not only non-linear objectives, but also non-linear equality and inequality constraints. This then allows general NLP (non-linear programming) problems to be reformulated as MIP (mixed-integer linear programming) problems.

**SOS2 constraints**: special ordered sets of type 2 (SOS2) constraints are normally used to model piecewise linear functions in convex and non-convex separable programming.

In the general case, an **SOS2 constraint** is completely defined by specifying a set of variables \( \{t_1, t_2, \ldots, t_n\} \) and this is equivalent to the following three constraints:

- \( t_1, t_2, \ldots, t_n \geq 0 \)
- \( t_1 + t_2 + \ldots + t_n = 1 \)
- only two adjacent variables, \( t_i \) and potentially \( t_{i+1} \), can be non-zero.

Given that we need to model the piecewise linear continuous function

\[
y = f(x)
\]

specified by its \( n \) node points \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\) as shown below.

The standard description using an SOS2 constraint is the following:

- \( x = x_1 t_1 + x_2 t_2 + \ldots + x_n t_n \)
- \( y = y_1 t_1 + y_2 t_2 + \ldots + y_n t_n \)
- **SOS2**: \( \{ t_1, t_2, \ldots, t_n \} \)

where the SOS2 variables \( t_1, t_2, \ldots, t_n \) play the role of interpolation parameters.

The implementation of SOS2 constraints within the simplex method assumes an additional rule to choose the variable to enter the basis. Namely, if \( t_j \) is basic, only \( t_{j-1} \) or \( t_{j+1} \) can be basic, while all other SOS2 variables have to be non-basic (and therefore fixed at zero).
However, since the set of feasible solutions may be non-convex, such a version of the simplex method enables only a local optimum to be obtained.

**Modeling piecewise linear functions in GLPK**  
SOS2 constraints are not implemented in GLPK, but a piecewise linear function can be easily modeled using binary variables as follows.

Let \( z_1, z_2, \ldots, z_{n-1} \) be binary variables, such that \( z_i \in \{0, 1\} \), where:

- \( z_i = 1 \) means that \( x_i \leq x \leq x_{i+1} \) and \( y_i \leq y \leq y_{i+1} \)

then, with \( s_1, s_2, \ldots, s_{n-1} \) segment variables, such that \( s_i \in \mathbb{R} \):

- \( z_1 + z_2 + \ldots + z_{n-1} = 1 \)
- \( 0 \leq s_i \leq z_i \) for \( i = 1, 2, \ldots, n-1 \)
- \( x = + x_1 z_1 + (x_2 - x_1) s_1 \\
  + x_2 z_2 + (x_3 - x_2) s_2 \\
  \ldots \\
  + x_{i-1} z_{i-1} + (x_i - x_{i-1}) s_{i-1} \\
  \ldots \\
  + x_{n-1} z_{n-1} + (x_n - x_{n-1}) s_{n-1} \)
- \( y = + y_1 z_1 + (y_2 - y_1) s_1 \\
  + y_2 z_2 + (y_3 - y_2) s_2 \\
  \ldots \\
  + y_{i-1} z_{i-1} + (y_i - y_{i-1}) s_{i-1} \\
  \ldots \\
  + y_{n-1} z_{n-1} + (y_n - y_{n-1}) s_{n-1} \)

The main advantage of this description is that the MIP solver is always able to find a global optimum.

**Modeling SOS2 constraints in GLPK:** if necessary, SOS2 constraints can be modeled independently when modeling a piecewise linear function thus.

Let \( \{t_1, t_2, \ldots, t_n\} \) be an SOS2 constraint. Then its equivalent description is the following:

- \( z_1 + z_2 + \ldots + z_{n-1} = 1 \)
- \( 0 \leq s_i \leq z_i \) for \( i = 1, 2, \ldots, n-1 \)
- \( t_1 = z_1 - s_1 \\
  t_2 = z_2 - s_2 + s_1 \\
  \ldots \\
  t_i = z_i - s_i + s_{i-1} \\
  \ldots \\
  t_{n-1} = z_{n-1} - s_{n-1} + s_{n-2} \\
  t_n = s_{n-1} \)

where \( z_1, z_2, \ldots, z_{n-1} \) are binary variables and \( z_i = 1 \) means that only \( t_i \) and potentially \( t_{i+1} \) are non-zero.